

AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated in the following listing of all claims:

1. (Original) A method of managing a generational memory, the method comprising:
sampling, at run-time of an execution sequence, lifetimes of a representative subset of
memory objects in the generational memory; and
pretenuring, based on the sampled lifetimes, at least some of the memory objects
allocated from the generational memory during the execution sequence.
2. (Original) The method of claim 1,
wherein the pretenuring is performed for those of the memory objects for which
corresponding sampled lifetimes exceed a first metric.
3. (Original) The method of claim 2,
wherein the pretenuring is reversed, if corresponding sampled lifetimes fall below a
second metric.
4. (Original) The method of claim 1,
wherein the pretenuring is performed while corresponding sampled lifetimes exceed a
metric.
5. (Original) The method of claim 1,
wherein the sampling includes sampling of representative subsets for plural categories of
the memory objects; and
further comprising:
allocating the memory objects using category-specific allocation functionality;
and
selectively modifying the category-specific allocation functionality to pretenure,
on subsequent allocations, memory objects corresponding to those of the
categories for which the sampled lifetimes exceed a metric.

6. (Original) The method of claim 5,
wherein the selective modification of the category-specific allocation functionality
includes instantiating category-specific allocation methods that allocate new
objects of a corresponding category directly into a tenured generation of the
generational memory.
7. (Original) The method of claim 1,
wherein the sampled lifetimes are characterized as per-category, mean lifetimes.
8. (Original) The method of claim 1,
wherein the sampled lifetimes are characterized as per-category, distributions of
lifetimes.
9. (Original) The method of claim 5,
wherein the categories are object class-specific.
10. (Original) The method of claim 5,
wherein the categories are call-site specific.
11. (Original) The method of claim 5,
wherein the categories correspond to activation record stack profiles.
12. (Original) The method of claim 5,
wherein the categories correspond to both type of memory object and call-site for
allocation thereof; and
wherein the selective modification of the category-specific allocation functionality
includes in-lining instructions at the corresponding call-site, the in-lined
instructions allocating new memory objects of the corresponding type directly
into a tenured generation of the generational memory.
13. (Original) The method of claim 1,

wherein the sampling includes sampling of representative subsets of the memory objects for plural categories thereof; and
wherein the pretenuring is performed on a category-specific basis.

14. (Original) The method of claim 1, further comprising:
establishing weak references to respective of the sampled memory objects and
associating allocation-time information therewith; and
identifying those of the sampled memory objects that become unreachable using the weak references.

15. (Original) The method of claim 14, wherein the weak references include one of:
phantom references;
references of strength less than any other reference by which an unreachable one of the
sampled objects may become reachable; and
references of strength less than any other reference employed in the computational system.

16. (Original) The method of claim 1, further comprising:
selecting the representative subset based on allocation buffer overflow.

17. (Original) The method of claim 1, further comprising:
selecting the representative subset using per class allocator functionality.

18. (Original) The method of claim 1, further comprising:
selecting the representative subset based on identity of an allocating thread.

19. (Original) The method of claim 1, further comprising:
coincident with allocation of memory objects of the representative subset, establishing
weak references thereto and associating therewith information indicative of at
least allocation time and call site.

20. (Original) A method of operating an automatically reclaimed storage environment in accordance with object lifetime statistics, the method comprising:

selecting representative subsets of memory objects for each of plural categories thereof; sampling, during a program execution, lifetimes of memory objects from the representative subsets; and tailoring, during the program execution, a storage management action based on the sampled lifetimes for a corresponding one of the categories.

21. (Original) The method of claim 20, wherein the storage management action includes pretenuring subsequently allocated memory objects of the corresponding category.

22. (Original) The method of claim 20, wherein the storage management action includes promoting memory objects of the corresponding category to a particular generation.

23. (Original) The method of claim 20, wherein the storage management action includes steering, on promotion, memory objects of the corresponding category to a particular store.

24. (Original) The method of claim 20, wherein the storage management action includes allocating subsequently allocated memory objects of the corresponding category from a particular store.

25. (Original) The method of claim 24, wherein the particular store is selected from the set of thread-local storage and global storage, based on the sampled lifetimes.

26. (Original) The method of claim 20, wherein the storage management action includes modifying a category-specific allocator in accordance with a storage management policy adapted to the sampled lifetimes; and

instantiating the modified, category-specific allocator.

27. (Original) The method of claim 20,
wherein the automatically reclaimed storage environment includes a generational garbage
collector; and
wherein the storage management action includes generation selection for unsampled
instances of the memory objects.

28. (Original) A storage management facility for a computational system, the storage
management facility comprising:
an object sampler operable to sample lifetimes of at least a subset of objects instantiated
in the computational system during execution of a program; and
a storage allocation facility operable during the execution of the program to allocate new
objects corresponding to respective of the sampled objects based at least in part
on the sampled object lifetimes.

29. (Original) The storage management facility of claim 28,
wherein the object sampler samples lifetimes on a per object category basis; and
wherein operation of the storage allocation facility is particular to each object category
and based at least in part on the lifetimes of the sampled objects corresponding
thereto.

30. (Original) The storage management facility of claim 28,
wherein the storage allocation facility includes category-specific allocators; and
wherein, in response to respective of the sampled object lifetimes exceeding a metric, the
category-specific allocators are modified at run-time to pretenure objects
allocated thereby.

31. (Original) The storage management facility of claim 28, wherein the categories
correspond to one or more of:
object type;
allocation call site;

activation record stack state;
thread id; and
receiver object.

32. (Original) The storage management facility of claim 28,
wherein the object sampler is responsive to transition of at least one of the sampled
objects from a reachable state to an unreachable state.

33. (Original) The storage management facility of claim 28,
wherein the object sampler employs a weak reference construct of the computational
system to identify those of the sampled objects that have become unreachable.

34. (Original) The storage management facility of claim 33,
wherein the weak reference construct includes a phantom reference.

35. (Original) The storage management facility of claim 33,
wherein the weak reference construct includes a virtual-machine-level weak reference of
strength less than any other weak reference by which an unreachable instance of
the sampled objects may become reachable.

36. (Original) The storage management facility of claim 33,
wherein the weak reference construct includes a virtual-machine-level weak reference of
strength less than any other weak reference employed in the computational
system.

37. (Original) The storage management facility of claim 28, further comprising:
a garbage collector,
wherein the object sampler is responsive to a storage management event of the garbage
collector affecting one or more of the sampled objects.

38. (Original) The storage management facility of claim 37,

wherein the storage management event includes collection of one or more of the sampled objects.

39. (Original) The storage management facility of claim 37, wherein the storage management event includes promotion of one or more of the sampled objects from a younger generation to an older generation.

40. (Original) The storage management facility of claim 37, embodied as a computer program product.

41. (Original) A computer program product encoded in at least one computer readable medium, the computer program product comprising:

at least one functional sequence for maintaining per-category object lifetime statistics based on a sampled subset of objects using weak references and associated allocation time information; and

at least one functional sequence for tenuring objects in accordance with those of the object lifetime statistics corresponding thereto.

42. (Original) A computer program product as recited in 41, wherein the tenuring sequence is instantiated for a particular category when corresponding object lifetime statistics exceed a metric.

43. (Original) A computer program product as recited in 42, wherein the tenuring sequence replaces a category-specific allocator.

44. (Original) A computer program product as recited in 42, wherein the tenuring sequence is replaced by a non-tenuring, category-specific allocator when corresponding object lifetime statistics fall below a second metric.

45. (Original) A computer program product as recited in 41,

wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.

46. (Original) An apparatus comprising:
means for sampling instances of software objects to maintain lifetime predictions for categories thereof; and
means for altering object category-specific storage management policies at run-time in response to the lifetime predictions.

47. (Original) The apparatus of claim 46, further comprising:
means for reversing the object category-specific storage management policies at run-time in response to the lifetime predictions.

48. (Previously Presented) The method of claim 1,
wherein at least some of the sampling is performed coincident with death of respective ones of the sampled memory objects.

49. (Previously Presented) The method of claim 20,
wherein at least some of the sampling is performed coincident with death of respective ones of the sampled memory objects.

50. (Previously Presented) The storage management facility of claim 28,
wherein at least some of the sampling thereby is performed coincident with death of respective ones of the sampled objects.

51. (Previously Presented) The computer program product of claim 41,
wherein the object lifetime statistics are based at least in part on sampling performed coincident with death of respective ones of the sampled objects.

52. (Previously Presented) The apparatus of claim 46,
wherein the sampling is performed coincident with death of the software objects